

**REMARKS**

Applicants respectfully request further examination and reconsideration in view of the above amendments and the comments set forth fully below. Claims 1-25 were pending. Within the Office Action, Claims 1-25 have been rejected. By the above amendment, Claims 1, 8, 15 and 21 have been amended. Accordingly, Claims 1-25 are now pending.

**Claim Objections**

Within the Office Action, the Claims have been objected to under 37 C.F.R. 1.52(b) because the lines are crowded too closely together. All the Claims now have a line spacing of 1.5.

Within the Office Action, Claims 15 and 21 have been objected to as reciting “the network of device” instead of “the network of devices.” By the above amendments, Claims 15 and 21 have been amended to state “the network of devices.” Therefore, Claims 15 and 21 are now in an allowable form.

**Rejections Under 35 U.S.C. § 112**

Within the Office Action, Claims 1-7 and 21-25 have been rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Office Action states that “it is vague and indefinite that the apparatus [or device] only comprises software means alone.” [Office Action, page 5]. By the above amendments, the independent Claims 1 and 21 have been amended to include the phrase “a memory comprising,” and therefore comprise more than software means alone. For at least this reason, Claims 1 and 21 are now in an allowable form.

Claims 6-7 and 22-25 are dependent on the independent Claims 1 and 21, respectively. As described above, Claims 1 and 21 are now in allowable form. Therefore, Claims 6-7 and 22-25 are all also allowable as being dependent on an allowable base claim.

**Rejections Under 35 U.S.C. § 101**

Within the Office Action, Claims 1-7 and 21-25 have been rejected under 35 U.S.C. §101 as the claimed invention is directed to non-statutory subject matter. Specifically, it is stated within the Office Action that the “claimed apparatus is software per se, and is therefore non-

statutory.” [Office Action, page 6]. By the above amendments, the independent Claims 1 and 21 have been amended to include the phrase “a memory device comprising,” and therefore comprise more than software means alone. For at least this reason, Claims 1 and 21 are now in an allowable form.

Claims 6-7 and 22-25 are dependent on independent Claims 1 and 21, respectively. As described above, Claims 1 and 21 are now in allowable form. Therefore, Claims 6-7 and 22-25 are all also allowable as being dependent on an allowable base claim.

### **Rejections Under 35 U.S.C. § 103**

Within the Office Action, Claims 1-13, 15-18 and 20-24 have been rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent Application Publication No. 2003/0182450 to Ong et al. (“Ong”) in view of SyncML Sync Protocol, version 1.0.1, [http://www.openmobilealliance.org/tech/affiliates/syncml/syncml\\_protocol\\_v101\\_20010615.pdf](http://www.openmobilealliance.org/tech/affiliates/syncml/syncml_protocol_v101_20010615.pdf) to Ericsson et al. (hereafter “Ericsson”). The Applicant respectfully disagrees.

Ong teaches a generic infrastructure for converting documents between formats with merge capabilities. The generic conversion framework allows developers to develop custom plug-in conversion algorithms and/or merge algorithms. [Ong, Abstract] Ong also teaches that the framework may provide a generic Application Programming Interface (API), through which one or more of the plug-ins may be plugged into the framework. The plug-in modules for converting, differencing and/or merging documents of various formats may interface with the framework via the API. [Ong, ¶ 0139] Ong further teaches that the framework is configurable to use any of a variety of **front-ends**, for example, email, HTTP, SyncML, WebDAV, SOAP and ebXML, among others. For example, in one embodiment with an email front-end, a user of a client device may email documents to a particular synchronization implementation based on the framework, which then may email the results (synchronized documents) back to the client device. [Ong, ¶ 0143] Ong does not teach *one or more synchronization protocol stacks*. Ong also does not teach that *the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks*.

Within the Office Action, it is stated that “Ong discloses a generic API that comprises a plurality of converting and merging plug-ins (for synchronization), wherein default plug-ins can be provided ... [and] that a plug-in is independent of an application [because] plug-ins can be

updated dynamically at runtime independently of an application.” [Office Action, page 3]. Based on this, the Office Action then concludes that “application communications with the API [are] independent of the synchronization *scheme* ... provided by the API.” [Office Action, page 3] (emphasis added). The Applicant respectfully disagrees.

It is stated within the Office Action that the application communications with the API are independent of the synchronization *scheme*. However, the claims of the present invention include a limitation specifying that the application communications with the API are independent of the synchronization *protocol*, not the synchronization *scheme*. Therefore, although the application communications of Ong may be independent of the particular configuration of the entire synchronization *scheme*, they are not independent of the synchronization *protocol* and therefore do not teach the present invention. The “generic API” disclosed by Ong is not the same as the generic API of the present invention. When Ong refers to the “generic API” it is referring to a framework (220) which includes *multiple APIs* that are each designed to communicate with *a single synchronization protocol* (not a protocol stack). In particular, Ong emphasizes this *multiple API* feature by teaching that the amount of APIs coupled to the framework (220) can be expanded through additional plug-ins in order to extend the “generic API’s” conversion capabilities. [Ong, ¶ 0140]. In other words, the “generic API” of Ong is a *collection* of individual APIs that need to be *called individually* depending on what synchronization protocol is desired. Accordingly, in the framework of Ong, each application communication with the “generic API” needs to be directed to and formatted for a particular API from the collection that is associated with the desired synchronization protocol. As a result, Ong teaches that each application communication to the API (or interface layer) *is dependent* on the particular synchronization protocol associated with that particular API because the communications must be *directed to and formatted for* the appropriate API that is associated with the desired synchronization protocol. The mere fact that “plug-ins can be updated dynamically at runtime independently of an application” in no way implies that the *format and direction* of the application communications to the API are independent of the desired synchronization protocol. Therefore, Ong does not teach *that messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks*.

Also within the Office Action, it is stated that “[t]he prior art does disclose one or more synchronization protocol stacks as a plurality of synchronization protocol such as HTTP,

SyncML, WebDAV, SOAP and ebXML (Ong, [0143]); whereas the application also uses SyncML, WebDAV as synchronization protocols.” [Office Action, page 2]. Based on this it is concluded within the Office Action that “[a]t least one synchronization protocols inherently exist in a protocol stack of the network device that supports the protocols.” [Office Action, page 2]. The Applicant respectfully disagrees. Ong only teaches that the API is configurable to use the ***front-end*** applications, not that the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the ***synchronization protocol stacks***. (Ong’s framework is ***configurable*** to use any of a variety of ***front-ends***, for example, email, HTTP, SyncML, WebDAV, SOAP and ebXML, among others. [Ong, ¶ 0143]). In other words, Ong teaches that ***one*** of the ***collection*** of APIs may be ***configured*** to receive (on the “***front-end***”) messages according to ***one*** of the particular protocols described. In fact, within the same paragraph Ong teaches that these front-end communications need to be sent “to a particular synchronization implementation.” [Ong, ¶ 0143]. This is very different than a single API that is able to choose from a plurality of synchronization protocols within a protocol stack. The APIs in Ong are each associated with particular synchronization protocols and therefore only reference a ***single*** synchronization protocol, not a ***set*** of synchronization protocols in a protocol stack. A single synchronization protocol cannot comprise an entire protocol stack and as a result, Ong does not teach a protocol stack. Ong teaches that the API is configurable to use the ***front-end*** applications, not that the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the ***synchronization protocol stacks***.

In contrast with the “generic API” of Ong, the present invention teaches a generic API that is referring to a ***single API*** that is capable of communicating to ***multiple different synchronization protocols*** (a synchronization protocol stack). The generic API of the present invention is not comprised of multiple APIs that need to be called individually, instead the present invention teaches a ***single*** generic API that ***all*** communications from the application can be directed to utilizing the ***same format*** regardless of the desired synchronization protocol. Further in contrast to the teachings of Ong, the generic API of the present invention provides an interface between synchronization applications and a plurality of synchronization protocols (which can form a protocol stack). The generic synchronization API acts as an abstraction layer for the plurality of synchronization protocols. A synchronization application interfaces with the

generic synchronization API in a manner *independent of the specific synchronization protocol*, and the generic synchronization API interfaces with each individual synchronization protocol. Unlike the present invention, Ong does not teach *messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks*.

Moreover, Ong teaches away from the problem that the present invention is designed to solve. The generic API of the present invention solves the complexity, time and cost for developing a synchronization application by providing a generic synchronization API, which appropriately formats the synchronization communication according to one of the conventional synchronization protocols, at least one of which is sufficient for successfully providing synchronization communications with another network device. [Present Specification, page 2, lines 30-31 and page 3, lines 1- 17] By contrast, Ong emphasizes that third-party vendors may develop custom plug-in modules that interface with the API to support custom synchronization tasks. In other words, Ong teaches developing a synchronization application by providing a collection of APIs that can **added to** with plug-ins over time in order to extend the synchronization application's abilities, which plainly teaches away from the present invention's solution of creating a single generic synchronization API that has **no need to ever be added to** with custom plug-in modules.

In summary, as described above, Ong teaches that the API is configurable to use the front-end applications. Ong does not teach that an interface layer communicates with the applications using a synchronization protocol independent of the protocol used between the interface layer and the synchronization layer. Furthermore, also as described above, Ong teaches away from the problem that the generic API of the present invention is designed to solve. Based on at least these reasons, Ong does not teach that *the generic synchronization communications between the one or more applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization layer*.

Within the Office Action, Ericsson is cited for teaching a synchronization layer which consists of synchronization protocols. However, Ericsson also does not teach that *the generic synchronization communications between the one or more applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization layer*. Accordingly, neither Ong, Ericsson nor their combination teach *the generic synchronization communications between the one or more applications and the interface*

*layer are independent of a synchronization protocol used between the interface layer and the synchronization layer.*

In contrast to the teachings of Ong, Ericsson and their combination, the generic API of the present invention provides an interface between synchronization applications and a plurality of synchronization protocols. The generic synchronization API acts as an abstraction layer for the plurality of synchronization protocols. A synchronization application interfaces with the generic synchronization API in a manner independent of the specific synchronization protocol, and the generic synchronization API interfaces with each individual synchronization protocol. The synchronization protocols can be a synchronization protocol stack in a synchronization layer. As described above, neither Ong, Ericsson nor their combination teach *the generic synchronization communications between the one or more applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization layer.*

The independent Claim 1 is directed to a first device to synchronize data with a second device. The first device of Claim 1 comprises a memory comprising one or more applications, a network layer coupled to interface with the second device, a synchronization layer coupled to the network layer to provide a synchronization protocol between the first device and the second device, and an interface layer coupled to communicate with the one or more applications and the synchronization layer to provide generic synchronization communications between the one or more applications and the synchronization layer, wherein the generic synchronization communications between the one or more applications and the interface layer are independent of the synchronization protocol used between the interface layer and the synchronization layer. As discussed above, neither Ong, Ericsson nor their combination teach *wherein the generic synchronization communications between the one or more applications and the interface layer are independent of the synchronization protocol used between the interface layer and the synchronization layer.* For at least these reasons, the independent Claim 1 is allowable over the teachings of Ong, Ericsson and their combination.

Claims 2-7 are dependent upon the independent Claim 1. As discussed above, the independent Claim 1 is allowable over the teachings of Ong, Ericsson and their combination. Accordingly, Claims 2-7 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 8 is directed to a network. The network of Claim 8 comprises one or more network devices, and an application device. The application device comprises one or more applications, a network layer coupled to interface with the one or more network devices, a synchronization layer coupled to the network layer to provide a synchronization protocol between the application device and the one or more network devices, and an interface layer coupled to communicate with the one or more applications and the synchronization layer to provide generic synchronization communications between the one or more applications and the synchronization layer, wherein the generic synchronization communications between the one or more applications and the interface layer are independent of any synchronization protocol used between the interface layer and the synchronization layer. As discussed above, neither Ong, Ericsson nor their combination teach *wherein the generic synchronization communications between the one or more applications and the interface layer are independent of any synchronization protocol used between the interface layer and the synchronization layer*. For at least these reasons, the independent Claim 8 is allowable over the teachings of Ong, Ericsson and their combination.

Claims 9-13 are dependent upon the independent Claim 8. As discussed above, the independent Claim 8 is allowable over the teachings of Ong, Ericsson and their combination. Accordingly, Claims 9-13 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 15 is directed to a method of providing an interface to one or more synchronization applications resident within a first device coupled to a network of devices. The method of Claim 15 comprises sending and receiving messages to and from the one or more synchronization applications through an interface layer to one or more synchronization protocol stacks, to synchronize data between the first device and at least one other device within the network of devices, wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks, and generating and receiving communications at the interface layer to complete data synchronization between the first device and the at least one other device within the network of devices. As discussed above, neither Ong, Ericsson nor their combination teach *wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks*. For at least these reasons, the independent Claim 15 is allowable over the teachings of Ong, Ericsson and their combination.

Claims 16-18 and 20 are dependent upon the independent Claim 15. As discussed above, the independent Claim 15 is allowable over the teachings of Ong, Ericsson and their combination. Accordingly, Claims 16-18 and 20 are all also allowable as being dependent upon an allowable base claim.

The independent Claim 21 is directed to an apparatus for providing an interface to one or more synchronization applications resident within a first device coupled to a network of devices. The apparatus of Claim 21 comprising a memory comprising means for sending and receiving messages to and from the one or more synchronization applications through an interface layer to one or more synchronization protocol stacks, to synchronize data between the first device and at least one other device within the network of devices, wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks, and means for generating and receiving communications at the interface layer to complete data synchronization between the first device and the at least one other device within the network of devices. As discussed above, neither Ong, Ericsson nor their combination teach *wherein the messages between the one or more synchronization applications and the interface layer are independent of a synchronization protocol used between the interface layer and the synchronization protocol stacks*. For at least these reasons, the independent Claim 21 is allowable over the teachings of Ong, Ericsson and their combination.

Claims 22-24 are dependent upon the independent Claim 21. As discussed above, the independent Claim 21 is allowable over the teachings of Ong, Ericsson and their combination. Accordingly, Claims 22-24 are all also allowable as being dependent upon an allowable base claim.

Within the Office Action, Claims 7, 14, 19 and 25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Ong in view of Ericsson and in further view of U.S. Patent Application No. 2003/0014483 to Stevenson et al. (hereafter “Stevenson”). Applicants respectfully disagree.

Claims 7, 14, 19 and 25 are dependent on the independent Claims 1, 8, 15 and 21, respectively. As described above, the independent Claims 1, 8, 15 and 21 are all allowable over Ong, Ericsson and their combination. Accordingly, Claims 7, 14, 19 and 25 are all also allowable as being dependent upon an allowable base claim.

PATENT  
Attorney Docket No.: SONY-26700

For at least the reasons given above, the Applicants respectfully submit that the claims are in a condition for allowance, and allowance at an early date would be appreciated. Should the Examiner have any questions or comments, the Examiner is encouraged to call the undersigned at (408) 530-9700 to discuss the same so that any outstanding issues can be expeditiously resolved.

Respectfully submitted,  
HAVERSTOCK & OWENS LLP

Dated: August 13, 2008

By: \_\_\_\_\_/Jonathan O. Owens/

Jonathan O. Owens

Reg. No. 37,902

Attorneys for Applicant